NPS52-80-007

NAVAL POSTGRADUATE SCHOOL

Monterey, California

DESIRABLE PROPERTIES OF A NETWORK TAXONOMY

Earl E. McCoy

Bernard J. Carey

May 1980

Approved for public release; distribution unlimited.

NAVAL POSTGRADUATE SCHOOL
Monterey, California

Rear Admiral J. J. Ekelund
Superintendent

Jack R. Borsting
Provost

The work reported herein was supported by the Department of Computer Science, Naval Postgraduate School, Monterey, California.

This report was prepared by:

for GORDON H. BRADLEY, Chairman
Dept. of Computer Science

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER NPS52-80-007	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Desirable Properties of a Network Taxonomy		5. TYPE OF REPORT & PERIOD COVERED Technical Report
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Earl E. McCoy Bernard J. Carey		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, CA 93940		12. REPORT DATE May 1980
		13. NUMBER OF PAGES 45
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Computer Networks, Taxonomy, Interconnection Mechanisms, Fault Tolerance, Hardware Primitives

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This paper reviews and analyzes the desirable properties of a computer network taxonomy from the point of view of its usefulness in a design procedure. A key factor that must be considered is that the design environment currently evolving uses functionally high level VLSI-based building blocks to construct various network architectures. This paper begins by reviewing the uses of a taxonomy in a network context, and continues with a review of specifications for network requirements. A set of hardware interconnection primitives is defined next. A review of the Anderson and Jensen taxonomy [Ander75] is then

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

S/N 0102-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered) presented, with a discussion of its completeness. The main thrust of this article is given in a section of attributes of a design oriented taxonomy. Finally, extensions are proposed for fault tolerant considerations and proto cols.

A computer network is defined to be a hetrogeneous collection of computer and the telecommunications subsystem linking them together. Here the proper ties of the various network architectures are of particular interest; the "user" computers (or processors) are considered as sources and sinks of messages being transmitted over the network. No distinction as to the geo- graphicla scope of the network is made because it does not impact the taxonomy considerations addressed here.

Desirable Properties of a
Network Taxonomy

by

Earl E. McCoy
and
Bernard J. Carey

Technical Report NPS52-80-007

Naval Postgraduate School
Monterey, California

May 1980

Abstract

This paper reviews and analyzes the desirable properties of a computer network taxonomy from the point of view of its usefulness in a design procedure. A key factor that must be considered is that the design environment currently evolving uses functionally high level VLSI-based building blocks to construct various network architectures. This paper begins by reviewing the uses of a taxonomy in a network context, and continues with a review of specifications for network requirements. A set of hardware interconnection primitives is defined next. A review of the Anderson and Jensen taxonomy [Ander75] is then presented, with a discussion of its completeness. The main thrust of this article is given in a section on attributes of a design oriented taxonomy. Finally, extensions are proposed for fault tolerant considerations and protocols.

A computer network is defined to be a hetrogeneous collection of computers and the telecommunications subsystem linking them together. Here the properties of the various network architectures are of particular interest; the "user" computers (or processors) are considered as sources and sinks of messages being transmitted over the network. No distinction as to the geographical scope of the network is made because it does not impact the taxonomy considerations addressed here.

Uses of a Taxonomy

The derivation of a meaningful taxonomy in any context is dependent upon the intended use of the classification scheme. If the resulting taxonomy is intended to succinctly convey certain attributes of the entities classified then the appropriate notation should no doubt be founded upon the most important attributes of the various entities. Typically the use of taxonomies is static in nature, that is, there is no particular emphasis upon the dynamics of the classification mechanism itself.

In some contexts the dynamics of the classification process is an important aspect of the taxonomy scheme. For example it may be important to quickly classify an entity into the correct class. This contrasts to the usual usage of a taxonomy (such as in biology) where the taxonomy is used only to infer the attributes of the entity based upon its classification. The former is a dynamic process involving decision making at several levels; the latter is a decoding process based upon the taxonomy notation.

Thus a taxonomy may be viewed as useful in two complementary ways: in one instance, given an unknown entity, classify it correctly by making a series of multivalued decisions based upon the entity's important (and discoverable) attributes; in the other instance, given a set of attributes of interest, discover the appropriate class of entities by making a series of multivalued decisions based upon the attributes. Here we use "attributes" to mean any property of the entity of interest; in the network taxonomy context example attributes are fault tolerance and communications topology. A design procedure could clearly benefit from the latter view of a taxonomy.

In particular, a correctly defined taxonomy will be useful, and even possibly essential, in a design procedure for translating a set of network requirements into the "best" network configuration satisfying those requirements. To be useful in this manner the designer must know how to measure each of the criteria used in the classification scheme, have available an objective function which combines the various attributes in a way appropriate to the network user's intentions and such that maximizing the functional value is tantamount to finding the "best" network.

In summary, a network taxonomy must be amenable to a sequence of multivalued decisions, each of which is based upon a measurable criteria appropriate to the user requirements, such that each decision stage successively prunes the solution space in an optimal way until a single, best network topology remains. Codification of this decision procedure would constitute the creation of a very useful and desirable design method. Unfortunately the knowledge is not presently available to permit a definitive design method; much research remains to be accomplished before any universally acceptable design procedure can be found.

This paper is primarily concerned with the derivation of a taxonomy useful in the manner described. To meet this goal, requirements important to the network user are reviewed, the specification of an objective function is reviewed, and then an analysis of a particular taxonomy is presented, extensions proposed, and conclusions drawn.

Network Requirement Specification

In this section the specification of network requirements is reviewed. An understanding of these requirements is necessary to the development of a taxonomy useful to a design method that translates those requirements for a given application into a best network topology.

Kaczmarek and McGregor [Kacz78] provide an excellent summary on the definition of the networking problem to be solved. They state that requirements are of two types: strategical, to provide scope and direction to the development of a solution space; and tactical, which governs the actual development of a solution. These types are categorized as qualitative and quantitative needs and desires, respectively.

Strategical requirements are classed as guidelines (necessary attributes of an acceptable solution), data processing factors (possible evolutionary patterns of communication carried), and issues and priorities (unresolved factors possibly impacting the strategy). Tactical requirements are the environment (location of user nodes), data movement requirements (message characteristics and timing), performance (level of service provided), and node interface requirements (user to network protocol). Judgement as to the necessity or completeness of these requirements is not made here; rather we accept these network requirements as a basis upon which to discuss the role of a network taxonomy scheme in a design method.

Suppose a user of a potential network has somehow generated a set of requirements of the type suggested above. The geographical locations of nodes are stated, the properties of the messages estimated (arrival rates, message lengths, source-destination statistics), and a user-to-network protocol has been established. Can a design method translate these system requirements into a best network topology? The answer is "no" because a measure of what constitutes "best" has not yet been provided. An "objective function" is needed that can be used to rank order the alternate network configurations by providing a single measure acceptable to the network user. The next section discusses the nature of an objective function in the network design context.

Objective Function Specification

The definition of the particular application for which a network is being designed must be complete enough to indicate the relative importance of the strategical and tactical requirements in a functional form. This function can be described as an "objective function" because it maps values of a set of independent variables (amount of each requirement currently provided) into a single dependent value. This dependent value is to be maximized (by definition), and hence describes the "objective" of the optimization process.

It is often the form of the objective function (and possibly of the constraints upon the independent variables) that provides a basis for an algorithm for deriving the optimal values of the independent variable; the linear programming problem is an example of this circumstance. It is unknown if the network design problem can be formulated in a manner such that an existing algorithm is applicable.

Some research on network measures has been accomplished. Gonzalez and Jordan [Gon79] have developed a framework for the quantitative evaluation of distributed computer systems. They define a dimensionless "figure of merit" as a weighted sum of the difference between the desired and actual amount of each of a set of attributes. They also present an analysis pertaining to the form of the weighting and to the effect of adding or deleting attributes. In particular they propose an approach that relates the figure of merit to a set of "functional primitives" that are common to all alternate designs. Examples of primitives are busses, processor and memory cycle time, communication protocols, and arbitration schemes. This seems to be a promising approach: what remains is to identify network attributes pertaining to the user requirements, the derivation principles for the attribute weights, and the definition of the 3}cO%=9al primitives -- hence only a framework is presented by Gonzalez and Jordan.

Several authors have proposed definitions of the independent variables that constitute the set of attributes needed for an objective function. Generally they consist of performance (a throughput measure), cost and place modularity, failure effect, switching complexity, reconfiguration potential (extensibility) degree of security or integrity, maintainability, and present value of system life cycle cost. These are discussed in detail in [Chou74] and [Grubb75], the latter being a definition of nine performance evaluation criteria recommended by the National Bureau of Standards. McGregor and Kaczmarek [McG78] describe in detail the criteria used in a network model used by the Network Analysis Corporation.

If an objective function includes a mechanism for mapping network attributes into functional primitives then the determination and definition of these primitives is an important task necessary to a network design method. In the next section a set of hardware interconnection primitives is defined.

The Hardware Interconnection Primitive Set

Because of the nature of digital hardware a basic set of hardware interconnection primitives (HIPs) may be defined. From this set any functionally representative computer network interconnection subsystem (ISS) can be constructed when the ISS is a primary mechanism which influences the operational attributes of a network [Carey79]. A later section discusses the ability of a particular taxonomy to adequately describe ISSs of interest.

Table 1 summarizes eleven functionally distinguishable HIPs needed to construct a functionally representative set of computer networks. Part (a) of Table 1 indicates those HIPs that may exist in serial or parallel versions. The second group (b) indicates three other HIPs that can be in any of several forms and group (c) indicates necessary but functionally passive HIPs that do not affect the architecture of a computer network.

More detailed information on these various HIPs is presented in a succinct form in Table 2. The exact physical implementation of these HIPs is unimportant here; rather we concentrate upon the function of each HIP in the computer network context. Each HIP embodies characteristics of and the functionality of hardware components used in existing computer networks; for example see [McCoy80].

Table 1. Basic set of HIPs from which the various Network architectures can be constructed.

(a) Type 1 HIPs having both serial and parallel versions

BIU	bus interface unit
LIU	loop interface unit
UAn	user adapter, n users
SWn	switch, any two of n ports
L	link

(b) Type 2 HIPs can be in any of several forms

CP	communications processor
BW	bus window
B	bus
M	memory

(c) Type 3 HIPs that are functionally passive

BR	bus repeater
BT	bus terminator

A number of specific ISSs may be examined within the context of the Anderson and Jensen [Ander75] classifications. Figure 1 shows a loop network (DDL in the AJ taxonomy) constructed from the LIU HIPs described above. Each ISS is presented using the PMS notation [Bell71]. Figure 1(a) shows a four node DDL network consisting of LIUs and links (the Ls). The unterminated lines projecting from the LIUs are ports to user node components not shown because they are not logically part of the network proper. Figure 1(b) illustrates how, in some network configurations, a UA4 HIP may be used to interface more than one user processor to the loop.

Table 2. Definition of Hardware Interconnection Primitives.

Type 1 HIPs

- BIU - Bus Interface Unit. Interfaces a serial/parallel link to a bus; the link side is assumed to conform to the bus protocol; minimal intelligence and buffering capability; typically interfaces a bus with a serial/parallel link; an UAn HIP, a SWn HIP, a BW HIP, or a CP HIP; failure does not effect the bus.
- LIU - Loop Interface Unit. Interfaces a serial/parallel link to a loop type architecture; there is enough buffering capability to store several incoming or outgoing packets; only a sending LIU can remove a packet from the loop; a receiving LIU marks a passing packet as "received", copies it into a buffer, and sends the packet on; capable of synchronizing itself with other LIUs; typically interfaces to an UAn or a CP HIP; failure disables the entire loop.
- UAn - User Adapter, n users. Interfaces n users to a serial/ parallel port; acts as a specialized switch; failure typically isolates the n users from the network.
- SWn - Switch, n ports. Connects any two of n serial/parallel ports for the duration of packet transmission; has enough intelligence to make a connection based upon destination address (based upon routing tables); some buffering capability; act as a specialized CP HIP; failure results in all links being blocked.

Figure 2 shows a completely interconnected star configuration call DDC in the AJ taxonomy. In Figure 2(a) a four node network is shown composed of SW4 HIPs. Again a user adapter HIP could be used to interface more than one user processor to a node if that were desirable. If a five node DDC network is to be constructed it could be made from "larger" switches, say a SW5, or it could be made from cascading together more than one "smaller" switch,

Table 2 continued
Type 2 HIPs

CP - Communications Processor. General intelligence to interface several serial/parallel ports to each other; requires a microprocessor; could be specialized via programming to perform a wide variety of functions; failure blocks all communication between the ports.

L - Link. Communications medium; contains no intelligence; may contain "boosters" to extend its effective length; may be serial/parallel; failure breaks communication between the end points of the link.

BW - Bus Window. Interfaces two internal busses when memory addresses indicate the necessity; in general allows the busses to operate independently; no buffering capability; failure isolates the two busses.

B - Bus. Implements the set of signal lines used by an interface system to which a multiplicity of devices are connected and over which messages are carried [IEEE75]; typically a higher bandwidth than a link; failure isolates all BIUs and stops all communication.

M - Memory. May be multiport; interfaces to a bus via a bus interface unit; failure effect depends upon the parity/error correction scheme used.

as shown in part (b). In the particular case one of the ports of the SW4s in not used because it is not needed. Figure 3 shows a shared memory or DSM network. Users would interface with the opposite ends of the links. Figure 4 shows a shared bus network with two bus interface units or BIUs; one has a user adapter attached. Figure 5 shows a single SW4 HIP used as the hub of a central star ICDS network. Again either a larger switch (ie, a SW5) or cascaded switches could be used for the construction of a ICDS network of more nodes.

Table 2 continued
Type 3 HIPs

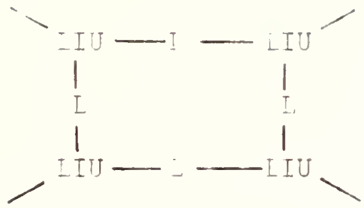
BR - Bus Repeater. Provides a boost in signal strength to allow a physical extension of a bus; failure results in the isolation of the bus components from each other.

BT - Bus Terminator. Prevents reflections from the ends of a bus; failure reduces the effective bandwidth; a passive

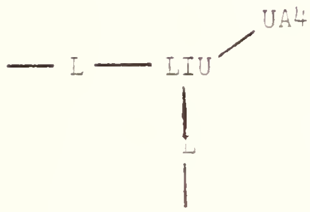
A loop with central switch is shown in Figure 6. Note the LIU HIP is used as in the DDL loop, but here the communications processor HIP (a CP) is employed to effectively produce an "intelligent LIU". Similarly Figure 7 shows a bus with central switch; a CP HIP is interfaced to the bus by a BIU. User processes would be attached to the other BIUs.

Figure 8 shows an example of a five user node irregular network (IDDI) composed from SW4s. Note three of the SW4 are underutilized. Figure 9 shows two direct shared bus networks (as in Figure 4) interconnected by a bus window HIP. Figure 10 shows an IDDR "regular" network made from SW5 HIPs. A variation of this using busses is shown in Figure 11; the MICRONET network [Witt78] is an example of an "IDSR" classification which was not included in the original AJ classification.

In this section we have defined a set of hardware interconnection primitives and exhibited their use in a variety of network configurations. The usefulness of these HIPs in a design

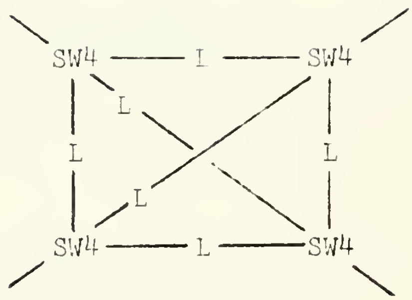


A) DDL network with four nodes

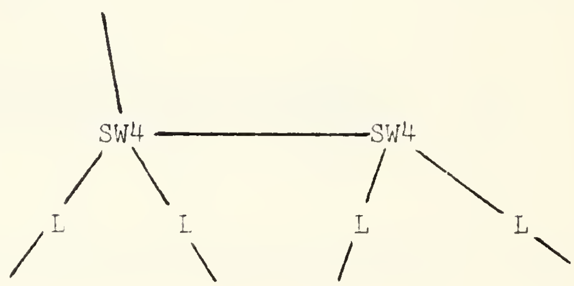


B) Detail of a single DDL node made to adapt up to four user processes by means of a UA4 HIP

Figure 1 A DDL network built from the basic set of HIPs.



A) Four node DDC "star" network built from completely utilized SW4 HIPs.



B) Detail of a single node of a five node DDC star network. Note how the SW4 HIP on the right is under-utilized.

Figure 2 Examples of DDC "star" networks built from the basic HIP set.

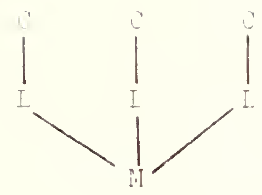


Figure 3 An example of a DSM network. User processes in the computers (Cs) interface with the links (Ls).

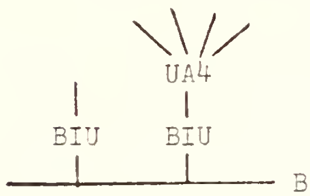


Figure 4 A DSB network with one single user node and another node supporting up to four users.

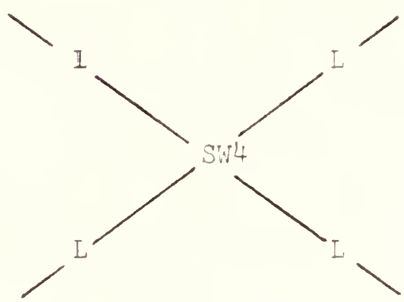


Figure 5 A ICDS central star network.

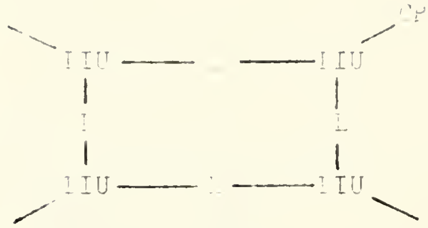


Figure 6 An ICDL "loop with central switch" network.

Figure 7 An ICSB "bus with central switch" network.

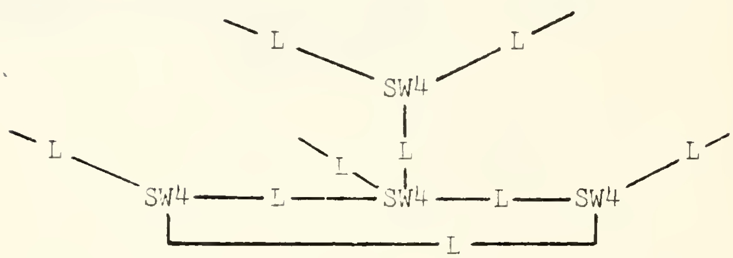


Figure 8 An IDDI "irregular network with six user nodes.

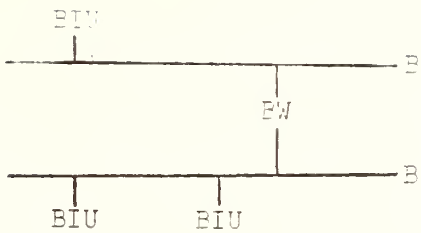


Figure 9 An IDS "bus window" network.

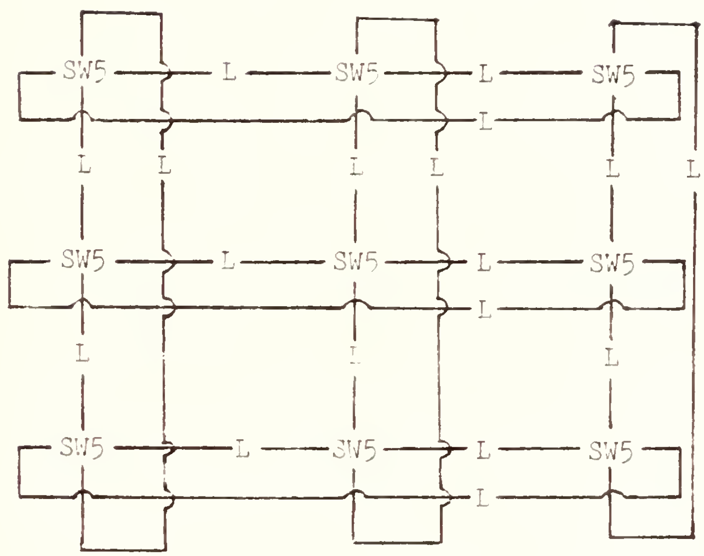


Figure 10 An IDDR "regular" network.

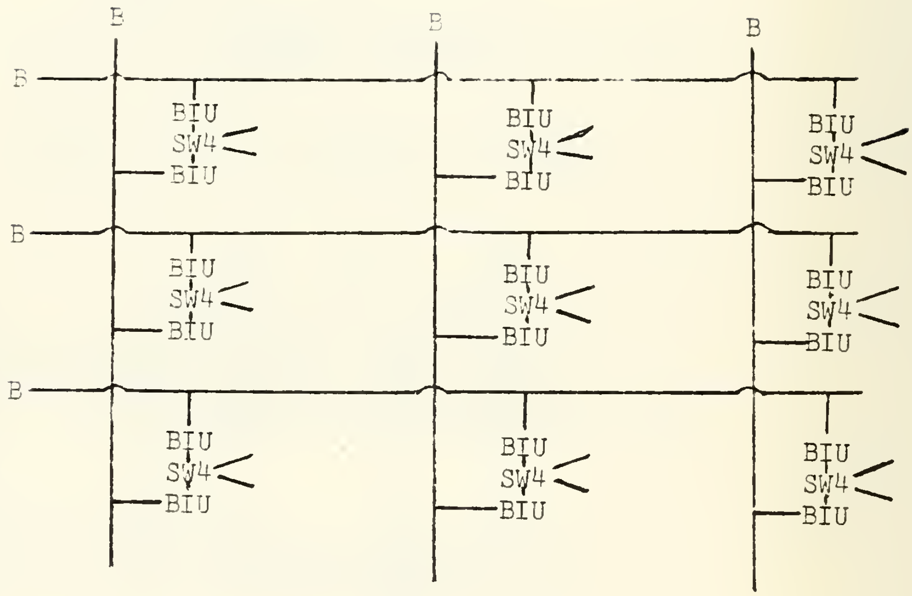


Figure 11 An IDSR "micronet" network.

method using the objective function ideas of Gonzalez and Jordan is unknown; the definition of a useful taxonomy is a prerequisite to an answer. In the next section the Anderson and Jensen taxonomy alluded to above is examined from this point of view.

The Anderson and Jensen Taxonomy

In this section the Anderson and Jensen taxonomy [Ander75] is reviewed because of its apparent usefulness as a base for classifying network architectures. It may also be extended to realize a more complete taxonomy upon which to base a network design methodology. Its underlying basis is examined and an analysis of its strengths and weaknesses concerning its potential role in an attribute/functional primitive driven design method is made.

Anderson and Jensen (AJ) view a network as a message passing medium with the hardware units forming the interconnection structure of a computer network as the basis of a taxonomy. In particular the hardware components of interest are paths and switches, as well as user nodes. A path is the medium over which a message packet is carried between processing nodes, and a switch is the intelligence along an indirect path between sender and receiver. Thus the hardware components are user processing nodes, paths, and switches.

A system architecture may then be characterized by the interconnection of these hardware components. AJ state that four levels or stages of decision making are adequate to classify the different ways in which the hardware components can be interconnected, and hence a tree structure can be used to represent the taxonomy. Figure 12 shows the AJ taxonomy tree. From the top level down the decisions concern message packet transfer strategy (direct or indirect), message packet transfer control method (none, centralized, or decentralized), transfer path structure (dedicated or shared), and finally a decision as to the the final network topology.

The usefulness of the AJ taxonomy in a network design procedure depends upon the ability to relate the four levels of decision to the original design requirements. Exactly how the design requirements translate directly or easily into a decision on, say, message packet transfer strategy is not immediately clear. It may be that other decisions can be made directly (and early on) from user requirements that have the identical effect of pruning the topological solution space.

Several computer network topologies do not fit smoothly into the AJ taxonomy. Various hybrids of the basic ten topologies can exist. These may be in the form of hierarchical networks such as RHEA [Pow78], or just coincidental networks interconnected by a "gateway" [Dav79]. As mentioned earlier the MICRONET network forms a new leaf in the AJ taxonomy tree which may be termed an

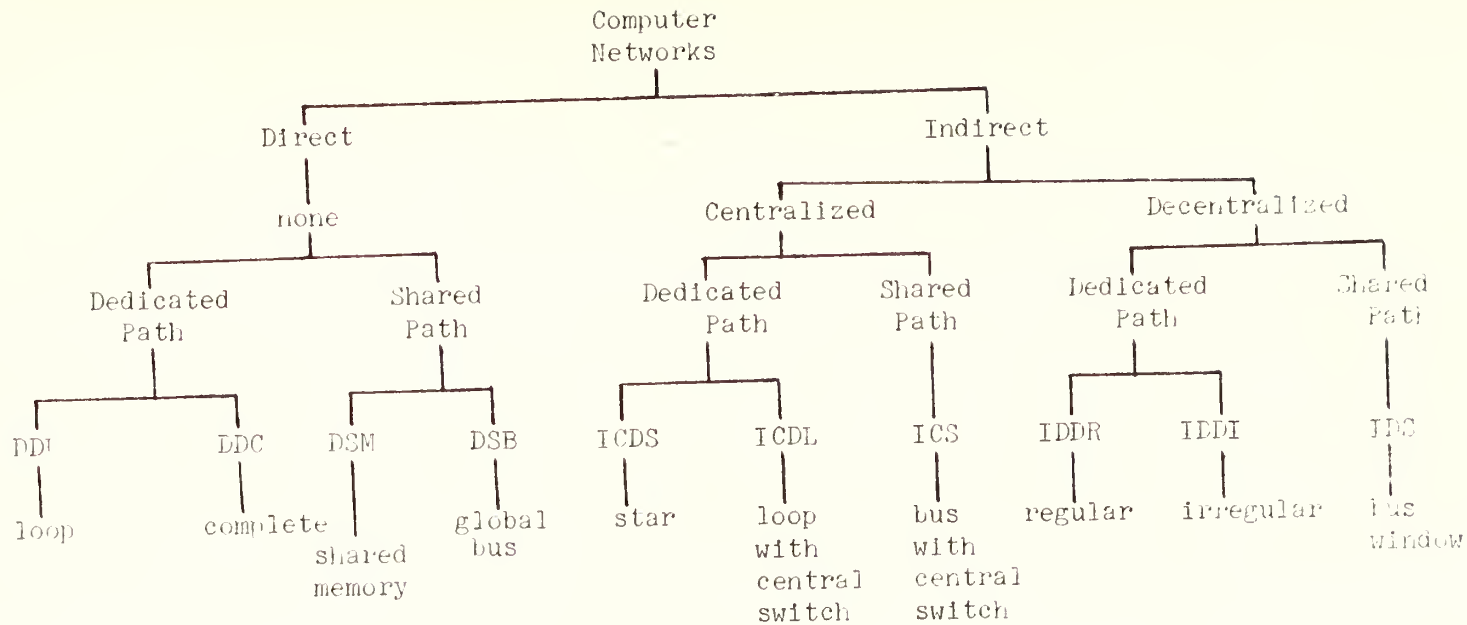


Figure 12 The Anderson and Jensen network taxonomy.

IDSR type network. Component redundancies for fault tolerance are not expressible in the AJ taxonomy, nor are any aspects of a network protocol.

In summary, the AJ taxonomy may be incomplete and even inappropriate for the design procedure environment but seems to provide the best conceptual structure at this time. The next section addresses those attributes necessary in a taxonomy from the design procedure viewpoint.

The Completeness of the AJ Taxonomy

In this section we examine the completeness of the AJ taxonomy with regard to the fundamentally unique network topologies. Recall that AJ define a "system architecture" level beneath three other layers of decision concerning transfer strategy, transfer control method, and transfer control structure. At the third level there exists three sets of dedicated paths and shared path pairs, the maximum provided by the decision alternatives allowed. From these six possibilities AJ define ten system architectures. It may be that other possibilities exist as was indicated by the Micronet system.

The first of the six transfer path structure possibilities is the DDx grouping that is subdivided into the DDL and DDC classifications. Clearly the DDL is the minimal way of connecting a given number of nodes in the DDx context and the DDC is the maximal way of connecting a set of nodes. It appears that adding more paths to the DDL network or removing some paths from the DDC network adds nothing significant in the way of system architecture. Such intermediate network types could be appropriately classified as an IDDI "irregular", hence this group is complete.

The second grouping is the DSx set, resulting in the DSM and DSB architectures. The distinguishing point here is the inclusion of memory as the path or not; note the bus behaves only as a temporary memory device. Here again the two subclasses are exhaustive, and so no new unique architecture can exist.

The ICDx grouping is next; it results in the ICDS "star" and the ICDL "loop with central switch" subtypes. By definition messages are sent to the switch and then retransmitted to their final destination. Within the ICD systems it seems that only the two possibilities can exist, and so this grouping is complete.

Only one architecture for the ICS grouping is defined by AJ. This "bus with central switch" architecture seems unique in that all the shared path types rely on a bus, and a centralized routing switch can be included in only one way; hence this class is complete. The definition of a switch in this classification seems too restrictive. A centralized bus arbitration scheme might be allowed in the ICS group, even though the message might be sent directly to the receiving node without retransmission. This appears to follow from AJ's definition of a switch as "the intervening intelligence between sender and receiver". Thus a polling scheme, such as used in SHINPADs [Kuhns79], might be permitted in the ICS classification.

The fifth group, IDDx, is subdivided into "regular" and "irregular", clearly an exhaustive set. Thus this group seems complete.

The last group is IDSx, consisting of the one classification IDS "bus window". In general this permits an irregular structure of busses. Since at least one "regular" network of the IDS type is described in the literature a second subclassification within this group should be defined, namely an "IDSR" type. The network described by this class is Wittie's MICRONET [Witt78]. To our knowledge this is the only non-hybrid network found in the literature requiring another system architecture within the AJ framework.

In summary, an analysis of the AJ taxonomy in terms of experimental networks indicates only one new system architecture exists, given the manner used to define the subclassifications; otherwise each group is divided into exhaustive classes, based upon some criterion such as interconnection, memory capability, a switch, regularity, or interconnection device type. We conclude that as far as basic classifications are concerned the AJ taxonomy is adequate, given the premise upon which it is based.

Attributes of a Design Oriented Taxonomy

In this section a basis for a network taxonomy amenable to an attribute/functional primitive driven design method is presented. The primary emphasis is placed upon the topological selection criteria; it may be that other considerations would influence the development of a taxonomy in other ways.

A useful taxonomy in a design environment should address the strategical aspects of a network topology critical to the network user. Examples of these aspects are the ability of a network to transmit the current and future message packet load; that it be maintainable and extensible, that it be fault tolerant to the degree desired; that it be place and cost modular with respect to the addition of future node sites; and that it be least expensive in terms of present value of life cycle cost. The main problem is to identify all the criteria that relate to topological decisions, relate them to user requirements, determine their relative importance to each other, and express the sequence of decisions in such a manner that the topological solution space is quickly pruned.

One approach to deriving such a taxonomy is to configure the possible computer networks given a set of MIPs, identify the resultant network attributes in relation to user requirements, and then to derive an appropriate sequence of decisions. This might be called a "bottom up" approach. A "top down" approach might be to identify the relationships between user requirements and decisions affecting network topology, making sure the decisions are answerable in terms of user requirements. The latter approach is more applicable to a design procedure, and so is pursued here.

The question is what user requirements relate to the highest level of network topological decision making. Clearly geographical considerations are one component in the highest level category. For example a network covering a large geographical area is most likely to be of an "irregular" nature just to keep interconnection costs reasonably low. However this need not be necessarily so: given appropriate traffic characteristics a radio medium based network (such as ALOHA [Abram70], or a satellite to user network) may be the least expensive in interconnection costs. Still, certain topologies might be excluded, such as loop or bus based networks, so geographical dispersement may be useful at a high level in a design procedure. If that is the case should the design decision be stated in terms of "message transfer strategy" (as it is in the AJ taxonomy) or some other statement closer to the requirements of the design problem?

Performance measures have the same type of problems. Given a performance measure (for example, messages per second) how does a designer infer a topological conclusion? The problem here is that any topology could theoretically be made to carry any message load, given appropriate technology. It may be that the functional primitives (HIPs in particular) could provide guidance in this area. The cost per bandwidth unit will be a step function in a building block design environment, hence matching performance to HIP may infer decisions about applicable ISSs and therefore network topologies.

The next section examines the possibility of extending the basic AJ taxonomy to include redundant HIPs for fault tolerance purposes.

Taxonomy Extensions for Fault Tolerance Considerations

An additional important consideration is the extension of the basic AJ taxonomy to include a notation to express the fault tolerant behavior resulting from the inclusion of HIP redundancies. This topic is specifically chosen because of its importance to network users and designers. As the AJ taxonomy now stands only the basic properties of a particular classification are described; additional HIPs that might be added for a specific purpose, such as fault tolerant reasons, have no mechanism permitting their description.

An example DSB architecture with a second, redundant bus is shown in Figure 13. Here the bus interface units (BIUs) are modified from the earlier definition such that two busses may be interfaced; note however, that they remain logically identical functional primitives. Each BIU must now contain the ability to compare the behavior of both busses and to determine which behavior is most likely to be correct. The point is that the fault tolerant ISS shown in Figure 13 remains a DSB architecture because of its overall behavioral attributes.

Equivalent situations can be easily formulated for the other classes of networks. DDL loop networks could be in "parallel" if suitable modifications were made to the LIUs. Any of the "regular" topologies could be modified in a similar manner. Some of the basic architectures imbed an equivalent level of fault tolerance without the need for additional, redundant components. For example, the MICRONET architecture mentioned earlier (a "regular" network of DSB architectures) already has the fault tolerant capabilities of the duplicate bus DSB described above. Hence the MICRONET "IDSR" architecture has inherent fault tolerance to some degree, and so does not necessarily require an extension to the taxonomy notation to express this fact. Thus the explicit reference to a duplicated HIP component does not seem to be a good way to express a level of fault tolerant capability.

Another approach to expressing a fault tolerant capability might be to determine the types of effect a single component HIP failure may cause. This may be called a "failure effect" way of describing fault tolerance, in contrast to a component oriented notation. For example, in the duplicated bus DSB configuration we could state that a failure of one of the two busses would cause no loss of communication between user nodes. Still, the failure of another HIP, say the BIU in this case, might isolate the user processes attached to it but not affect the communication between the remaining user processes. Thus at least two major failure effect modes, loss of node and loss of network, must be resolved. This implies that the particular type of component HIP that fails must be taken into consideration in a useful notation.

Given the situation described above it seems appropriate to define classes of fault tolerance suitable to describe the effect of its failure. Thus positional notation could be used to indicate the type of HIP failure, and encodings in each position could be defined to indicate its failure effect. A two component encoding is therefore proposed, both of which indicate a failure effect of a class of HIPs. The first component of the pair relates to the failure of an interface HIP, and the second component of the pair to the failure of a communication path HIP (a link or switch). Encodings for the effect must be descriptive, succinct, and easily remembered; we suggest "T" for tolerant, "L" for localized, and "V" for vulnerable. Table 3 indicates the definition of these encoding in more detail.

Certainly more detail could be included into the notation but it

Table 3. Effect Encodings.

Encoding	Failure Effect
T	Tolerant fault effect behavior. Failure means no loss of network capability.
L	Localized fault effect behavior. Failure means only locally attached user processes are isolated from the remaining processes.
V	Vulnerable fault effect behavior. Failure means the entire network becomes inoperative.

would be of little additional value to a reader interested in the particular fault tolerant behavior of the network; other notational schemes, such as a graphical representation of the network, could provide implementation details to a reader, if additional detail were needed.

The notation proposed is the use of the PMS notation of Bell and Newell [Bell71] in which attributes of a system are enclosed in square brackets. In this context the first entry is the AJ classification code, the second entry codes the effect of an interface HIP failure and the third codes the effect of a path HIP failure:

NETWORK := [<class code>;<failure code>;<failure code>]

where each <failure code> is a "T", "L", or a "V".

The worth of the proposed extension to the basic AJ classification scheme can best be demonstrated by some examples.

Consider the duplicated bus DSB architecture of Figure 13; in the proposed notation it could be described as simply DSB (if its fault tolerant behavior was not important) or as [DSB;L;T]. The "L" refers to the local effect of a BIU failure and the "T" refers to the tolerant behavior in the face of a single bus failure. An ordinary DSB architecture (without a duplicated bus) could be classified as a [DSB;L;V]. Similarly there could exist a [DDL;L;T]. Figure 14 shows an obvious configuration for a [DDL;L;T] and Figure 15 shows another equivalent version. The notation proposed here does not distinguish between the two versions (because their fault tolerant behavior is identical), hence implementation details are not explicitly indicated.

MICRONET could be classified as an IDSR or as an [IDSR;L;T] without regard to the presence of HIP redundancies. Similarly a DDC "complete" architecture could be classified as a [DDC;L;T] without HIP redundancies. The situation of the IDDI "irregular" networks in not so clear. For user nodes connected in a minimum spanning tree manner (fewest number of links possible) a path failure could isolate (in general) more than one node's user processes from the others and hence would be an [IDDI;L;L]. If more paths existed within the irregular network it could be of the class [IDDI;L;T]. Thus in the IDDI case redundant HIPs have a variable effect on fault tolerance depending upon the number and placement of the redundancies.

We have shown how path HIP components may be configured such that a network may require a T, L, or V encoding. In contrast all the examples shown have had the first "interface failure effect" code a "L" for a localized effect. In general the code can become a "T" only when an interface unit is duplicated and the same user processes connected to both interface components. The resulting situation is that two nodes now exist where only one existed before, and so a larger network results. Hence the fault tolerance capability is "outside" the network proper and need not be explicitly shown. The interface coding can become a "V" only when a unit (say a LIU) failure blocks all communication in the network.

In summary the notation proposed here is useful in describing the fault tolerant effect of interface unit failures and path failures. Three levels of effect are provided for each type of failure. More detail is considered to be of little practical use and would result in a more complicated encoding scheme than its worth. The value of the notation scheme proposed has been demonstrated by several examples.

Taxonomy Extensions for Protocols

A useful taxonomy classification scheme should have provisions for the inclusion of the description of an appropriate level of protocol because it conveys the type of user terminal equipment that could be attached and something about the performance characteristics of the network. This section reviews the problems associated with the development of such a notation and make a recommendation for a particular scheme.

A communications protocol is defined as those conventions necessary for the proper transmission of messages over a network. Typically, several layers of protocol are defined corresponding to the various functional needs involved. An advantage of considering layers of protocols is that each functional layer can be made essentially independent of the other layers, such that changes in any particular layer need not affect the others. Several definitions of the various layers exist in the literature and are germane here. For purposes of discussion the International Standards Organization (ISO) model of protocol layers is shown in Table 4 [ISO].

Table 4. ISO Protocol Level Model.

Level number	Function
7	Process control.
6	Presentation control.
5	Session control.
(transport mechanism)	
4	Transport end-to-end control.
3	Network control.
2	Link control.
1	Physical control.

Of importance here is what constitutes the appropriate level for inclusion in a useful taxonomy scheme. The two basic choices are at the "host-to-host" level or the physical/link level.

Consider the "host-to-host" level; this is the level that is "seen" by the network user. In the ISO model shown in Table 4 the user sees a combination of levels 4 and 5, which are concerned with both sides of the transport medium barrier. Walden and McKenzie [Wald79] point out this fact as an indication of the possible inappropriateness of the ISO model. Some other host-to-host protocols have been defined: examples are the Department of Defense (the Arpanet TCP protocol and the Autodin II protocol), the Consultative Committee for International Telegraphy and Telephony (CCITT) X.25 (includes a physical link level X.21, a logical link that is a subset of the HDLC protocol, and a packet level interface protocol), as well as various computer vendors such as IBM, DEC, Prime, Burroughs, etc. Much international effort at standardization is underway to define a true international standard but events (like a de facto standard) could overcome them and render them moot.

An alternate approach may be to concentrate upon the physical control level of protocols only, leaving the higher level of interfacing unstated. The problem with this is that even this level is not resolved as to standardization. Still the issues are not as volatile and at least one acceptable protocol is widely used even now. This is the EIA RS-232 protocol for which many terminal units are manufactured. Example of other protocols are X.24, X.26 (RS-423), X.27 (RS-422), and DIS 4093; see [Folts79] for a discussion of these protocols.

Another standardization effort is currently underway by the IEEE (see March 27,1980 issue of Electronics, p. 40). This effort is directed specifically to local computer networks; however, they should have significant ramifications to computer networks in general.

At this time it seems appropriate to define a notational scheme for the host-to-host level in spite of the fluid state of affairs at this level. We make this decision solely upon its usefulness to the potential network user and to the network designer. This follows particularly from the fact that a protocol at this level usually implies a physical level protocol as well, although it need not do so.

Considering protocols at this level as part of a taxonomy classification scheme represents some risk because of the standardization effort versus the manufactures rush to market a particular vendor unique system. Even so we make a recommendation in this respect. The particular nature of the recommendation follows the format of the optional fault tolerant notation described in the preceding section. As before the encodings should be succinct and meaningful. Table 5 lists some of the protocols currently in use; other most likely exist. At this time no attempt is made to encode each protocol type. Instead, until several de facto or true standards come into prominence we suggest that the full names shown in Table 5 be used. The list of networks is adapted from [Free79].

Table 5. Protocol Types.

Cm*	EPIC-DPS
LCS	TECHNEC
ICS	SHINPADS
SPIDER	MISS
FIBERNET	IRCN
MININET	DNC
DATARING	MITRENET
RIT NETWORK	ISUnet
DDN	KUIPNET
C.mmp	PLURIBUS
AN/USQ-67	ARGONNE
DCS	CYBERNET
DLCN	ETHERNET
DDLCN	NBS
BATNET	PRIMENET
HYPERCHANNEL	RIG
LASL	CERN
LABOLINK	OCTOPUT
X.25	

For example a particular network could be classified as a [DSB;ETHERNET]. A fault tolerance field could also be appended: [DSB;L;T;ETHERNET].

In summary we recommend that an optional appendage indicating the type of host-to-host protocol be made a part of a taxonomy scheme. Its presence would convey important information to the reader, and would be useful to a future design method. We choose to use the commonly accepted notations for the various networks currently used until that time true internationally recognized protocols at this level are

Summary

We have reviewed the attributes of a network taxonomy for a design procedure context. Among the conclusions drawn is a determination that the Anderson and Jensen taxonomy is sufficient for characterizing the high level structures of networks and appears to be useful as a base upon which to define extensions that encompass implementation considerations, fault tolerant attributes, and communication protocols. Particular extensions are proposed that seem advantageous in the high level functional primitive building block design environment. The next area that must be studied is the objective function area before a good design method for networks can be devised. To some extent work on protocol descriptions is dependent upon international and national standardization efforts, in addition to research into protocols themselves. In summary we believe the taxonomy extensions proposed here should prove to be of use in a future design procedure for the computer network context.

BIBLIOGRAPHY

[Abram70] Abramson, N., "The ALOHA System - Another Alternative for Computer Communications", FJCC, AFIPS Conf. Proc., Vol. 37, 1970, pp 281-285.

[Ander75] Anderson, G. A., and E. D. Jensen, "Computer Interconnection Structures: Taxonomy, Characteristics, and Examples", Computer Surveys, Vol. 7, No. 4, Dec 1975, pp 197-213.

[Bell71] Bell, C. G., and A. Newell, "ISP and PMS Descriptive Systems", Chapter 2 in Computer Structures: Readings and Examples, McGraw-Hill, 1971.

[Carey79] Carey, B. J., and E. E. McCoy, "Interconnection Subsystems for Multiple Microcomputers", Thirteenth Annual Asilomar Conference on Circuits, Systems, and Computers, Asilomar, Ca., November 1979.

[Chou74] Chou, W., "Planning and Design of Data Communications Networks", Proc NCC, Vol. 43, 1974, AFIPS Press, pp 553-559.

[Dav79] Davies, D. W., et. al., Computer Networks and Their Protocols, Wiley, 1979.

[Folts79] Folts, H. C., "Status Report on New Standards for DTE/DCE Interface Protocols", Computer, Vol. 12, No. 9, Sept 1979, pp 12-19.

[Free79] Freeman, H. A., and K. J. Thurber, "Issues in Local Computer Networks", ICC'79 Conf. Rec., pp 20.3.1-20.3.5.

[Grubb75] Grubb, D. S., and I. W. Cotton, "Criteria for the Performance Evaluation of Data Communications Services for Computer Networks", NBS Tech. Note 882, Sept 1975.

[Gon79] Gonzalez, M. J., and B. W. Jordan, Jr., "A Framework for the Quantitative Evaluation of Distributed Systems", Proc 1st Int'l Conf on Distributed Computing Symposium, Huntsville, Ala., Oct 1-5, 1979, pp 156-165.

[IEEE75] IEEE Standard Digital Interface for Programmable Instrumentation, IEEE Std 488-1975.

[ISO] ISO/TC 97/SC 16 N34, "Provisional Model of Open Systems Architecture", Computer Communications Review, Vol. 8, No. 3, July 1978, pp 49-61.

[Kacz78] Kaczmarek, R., and P. V. McGregor, "The Synthesis of Formal User Requirements: Defining the Networking Problem to be Solved", Proc. COMPCON 78, Sept 5-8, 1978, pp 411-418.

[Kuhns79] Kuhns, R. C., and M. C. Shoquist, "A Serial Data Bus System for Local Processing Networks", Spring

COMPCON 79, pp 266-271.

[McCoy80] McCoy, E. E., and B. J. Carey, 'An Examination of Representative Local Computer Networks, Their Architectural Classes, and Their Hardware Interconnection Devices', Proc. MIMI-80, Asilomar, Ca., Jan 1980.

[McG78] McGregor, P. V., and R Kaczmarek, "Modeling Network Architectures", Proc. Fall COMPCON 78, Sept 5-8, 1978, pp 419-426.

[Pow78] Powell, D., et. al., "RHEA: A System For Reliable and Survivable Interconnection of Real-Time Processing Elements', Proc. 1978 Int'l Sym. on Fault-Tolerant Computing, Toulouse, France, June 1978.

[Wald79] Walden, D. C., "The Evolution of Host-to-Host Protocol Technology", Computer, Vol. 12, No. 9, Sept 1979, pp 29-38.

[Witt78] Wittie, L. D., "MICRONET: A Reconfigurable Microcomputer Network for Distributed Computer Research", Simulation, Nov. 1978, pp 145-153.

INITIAL DISTRIBUTION LIST

Defense Documentation Center Cameron Station Alexandria, VA 22214	2
Library Code 0142 Naval Postgraduate School Monterey, CA 93940	2
Office of Research Administration Code 012A Naval Postgraduate School Monterey, CA 93940	1
Professor Bernard J. Carey Code 52Ck Naval Postgraduate School Monterey, CA 93940	10
Professor Earl E. McCoy Code 52My Naval Postgraduate School Monterey, CA 93940	10
Professor G. H. Bradley Chairman Department of Computer Science Naval Postgraduate School Monterey, CA 93940	20
Professor D. E. Kirk Chairman Department of Electrical Engineering Naval Postgraduate School Monterey, CA 93940	2